Title	FSD350v5 – SimpleCAN protocol specification and requirements
Products	SimpleCAN
Requirements	EN 61784-3:2021, EN 61508 series, EN ISO 13849-1
Purpose	Describe the SimpleCAN protocol and safety requirements.

Revision history

Version	Changes	Changed by	Date
V1	Initial version.	WF	2024-01-31
V2	• Update CRC structures for optimization • Define “node hash” and use in CRC • Specify endianness of values	WF	2024-03-15
V3	• Added document number • Added header • Restructured according to 61784-3:2021 annex C. • Updated definitions. • Updated calculations with FSD351 • Clarified purpose under scope	WF	2024-04-27
V4	• Correct memory packet diagram to use 6 bit timestamps • Added numbers for all figures. • other minor corrections.	WF	2024-05-10
V5	• Updated header product field and scope to only specify SimpleCAN.		

Introduction

This document specifies a safety-related communication layer (SCL) using the profile defined in ISO 11989 (Controller Area Network, CAN 2.0A/B). Many concepts described in this document refer to EN 50325-4 (CANOpen) and EN 50325-5 (CANOpen Safety), however, the protocol itself (SimpleCAN) is not based on these protocols.

1 Scope

This document applies to networks based on ISO 11989-1 providing safety-related communication capabilities between devices in a safety-related system in accordance with the requirements of EN 61508 series for functional safety.

The purpose of SimpleCAN is to enable the possibility to connect different Safety Simplifier networks to share safety-related data between them, and to enable the possibility to share safety-related data with other systems (such as Siemens/ProfiNET) via a gateway.

2 Normative references

- EN 61508 series;
- EN 61784-3:2021;
- EN ISO 13849-1;
- ISO 11989-1;

3 Terms, definitions, symbols, abbreviated terms and conventions

For the purposes of this document, the following terms and definitions apply.

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in EN 61784–3, 50325-5 and the following apply.

3.1.1 SimpleCAN

The protocol defined in this document.

3.1.2 Packet

A CAN-frame containing all fields specified in ISO 11989-1. Used interchangeably with **telegram** or **frame**.

3.1.3 Global time

Absolute time stamp for the network as defined in EN 61784-3.

3.1.4 Active master

The SCL currently acting as master on the bus.

3.1.5 Non master

All SCLs connected to the bus that are not acting as master.

3.1.6 Node hash

The unique hash used to identify SR data.

3.1.7 Cycle

A network cycle where all producers send their data once.

3.1.8 Slot

A millisecond during a cycle.

3.2 Symbols and abbreviated terms

For the purposes of this document, the following abbreviations apply.

3.2.1 Common symbols

- **SR**: Safety Related
- **NSR**: Non Safety Related
- **SRCP**: Safety Related Communication Protocol
- **SCL**: Safety Communication Layer
- **SRLD**: Safety Related Logical Device
- **CRC**: Cyclic Redundancy Check
- **DLL**: Data Link Layer

3.2.2 Additional symbols

- **HW**: Hardware
- **SW**: Software
- **SC-ID**: SimpleCAN ID
- **SDD**: Safety Data Dictionary

3.3 Conventions

The structure of this document follows the structure defined in EN 61784-3:2021 Annex C. “Mandatory” categorizes functionalities that shall be used or implemented; “optional” categorizes functionalities that may be used or implemented.

4 Overview of SimpleCAN

SimpleCAN is based on ISO 11898-1. It employs a multicast (EN 50325-1:2002) model similar to the producer/consumer push model (defined in EN 50325-4:2003 subclause 4.4.4), where a producer periodically transmits safety data and consumers receive the data. Physical nodes in the network (SRLD) can act as producers and consumers. An individual SRLD may be a producer of several messages/SC-IDs, and consumers may receive and handle multiple messages/SC-IDs. The SRLD communicates with the SCL by means of a safety-related data dictionary (SDD), which maps SC-IDs to SR data.

Safety data and non safety data can coexist on the same physical medium. Measures are employed to ensure NSR data cannot influence any safety function.

The safety data transfer is executed as follows:

1. The producing SRLD updates the SDD with the SR data to be sent.
2. The producer reads the SDD and packs the SR data and transmits it on the bus.
3. Consumers of the data receive the data, verify the CRC and update the SDD with the SR data and calculated age.
4. The consuming SRLD reads the SR data from the SDD.

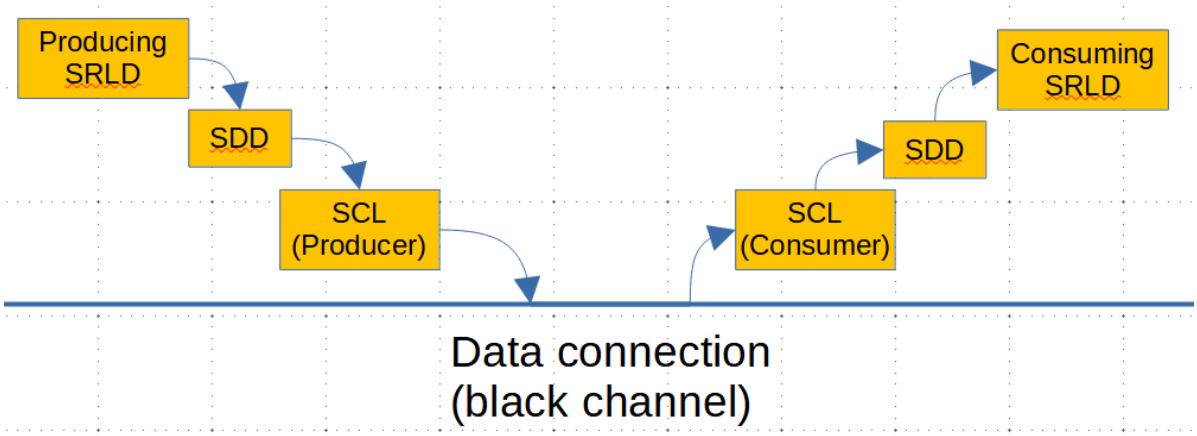

Figure 1: Flow of SR data. Note that the SCL and SDD exist twice in each SRLD.

To ensure the integrity of safety messaging and protect against all communication errors specified in IEC 61784-3:2021 chapter 5.3, SimpleCAN employs the following measures:

- Time stamp;
- Time expectation;
- Connection authentication;
- Data integrity assurance;
- Different data integrity assurance systems.

SR data is sent redundantly and cyclically. Diverse measures for producing SR messages are used to ensure that NSR messages are not interpreted as SR messages.

5 General

5.1 External documents providing specifications for the profile

Concepts defined in the following standards are referenced in this document and are useful in understanding the design of SimpleCAN:

- EN 61784-3:2021;
- ISO 11898-1:2015 (CAN, Controller area network);
- EN 50325-4:2003 (CANOpen);
- EN 50325-5:2010 (CANOpen Safety).

Other standards useful in understanding the principles of safety related systems:

- EN 61508:2010 series;
- EN ISO 13849-1.

5.2 Safety functional requirements

- The SRCP defined in this document (SimpleCAN) shall fulfill the requirements to be able to support SIL3 (according to EN 61508 series) and up to category 4 (according to EN ISO 13849-1).
- SimpleCAN shall only be used with devices operating in high demand continuous mode.
- The SRCP shall contribute to at most 1% of the maximum PFH or $PFHD_{avg}$ of SIL3.
- The safe state for digital and analog values shall be defined as 0.
- The SRCP is implemented using the black channel approach.
- Implementations of this SRCP shall comply with EN 61508 series.
- Safety devices shall comply with the increased test levels and durations, as well as corresponding performance criteria specified in IEC 61326-3-1 or the generic standard IEC 61000-6-7.
- SR communication shall be independent from NSR communication. However, NSR communication may use SR communication for transmission.

5.3 Safety measures

The employed safety measures and which communication errors they protect against are shown in the table below:

Communication errors	Safety measures				
	Time stamp	Time expectation	Connection authentication	Data integrity assurance	Different integrity assurance systems
Corruption				X	
Unintended repetition	X				
Incorrect sequence	X				
Loss		X			
Unacceptable delay	X	X			
Insertion	X		X		
Masquerade			X		X
Addressing			X		

Notes:

1: The safe state of values being 0 in combination with time expectation allows the consumer to safely treat timeouts/loss as safe state. Thus, time expectation is effective in handling lost packets in consumers.

No acknowledgment of SR data shall be used. Producers shall not implement any safety function that depend on successful reception in consumers.

5.3.1 Safety measures against possible communication errors

- Corruption:
  Data integrity assurance (5.4.7): SR data is protected by a 24 bit CRC.
- Unintended repetition:
  Protected by time stamps (5.4.3): The 24 bit CRC is based on the time stamp of both SCLs, new packets are guaranteed to be different with a new timestamp. The timestamp is 40 bits counting milliseconds, so it will not wrap around to 0 until after over 34 years. Identical packets have no effect: if the same message is received

multiple times, since the time stamp is known, the receiver calculates the same age as the previous message, and the result is the same.

- Incorrect sequence:
  Protected by time stamps (5.4.3): messages from a node which are older than the newest received message from that node can be discarded.
- Loss:
  Lost packets are handled as timeouts. If a timeout occurs the related SR data is set to safe state (0).
- Unacceptable delay:
  Protected by time stamps (5.4.3) and clock synchronization (8.1.2).
- Insertion:
  Protected by connection authentication (5.5.6): All SimpleCAN messages contain a 24 bit CRC that is generated using parameters only known to the SCLs participating in the SimpleCAN protocol, which ensures all recipients can guarantee the packet source is the correct one.
  Also see Addressing below.
- Masquerade:
  All safety related data is protected by a 24bit CRC, which can only be calculated by known network participants.
- Addressing:
  Protected by connection authentication (5.5.6): Since CAN is a broadcast medium, all nodes receive all packets that are sent on the bus, even if they are not the intended recipient. During reception, the receiver discards the packet if the CRC is invalid, or if the source ID is not configured as a source in the receiver.

5.4 SCL structure

Only one field-bus shall be used as the communication channel. From the models considered in 61784-3:2021 annex A, Model A (A.2) shall be used for transmission. For reception, model A or model C shall be used.

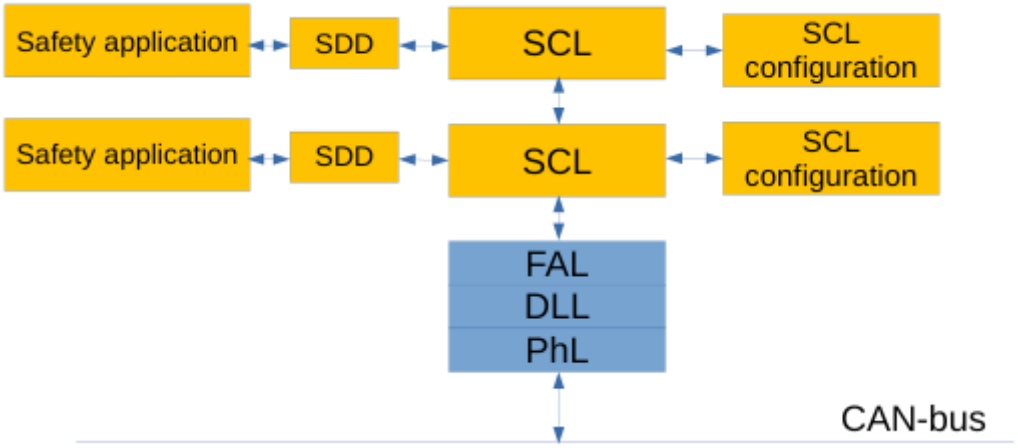

Figure 2: Example SCL structure using 61784-3:2021 Annex A Model A (A.2). SR layers are shown in orange and NSR layers are shown in blue.

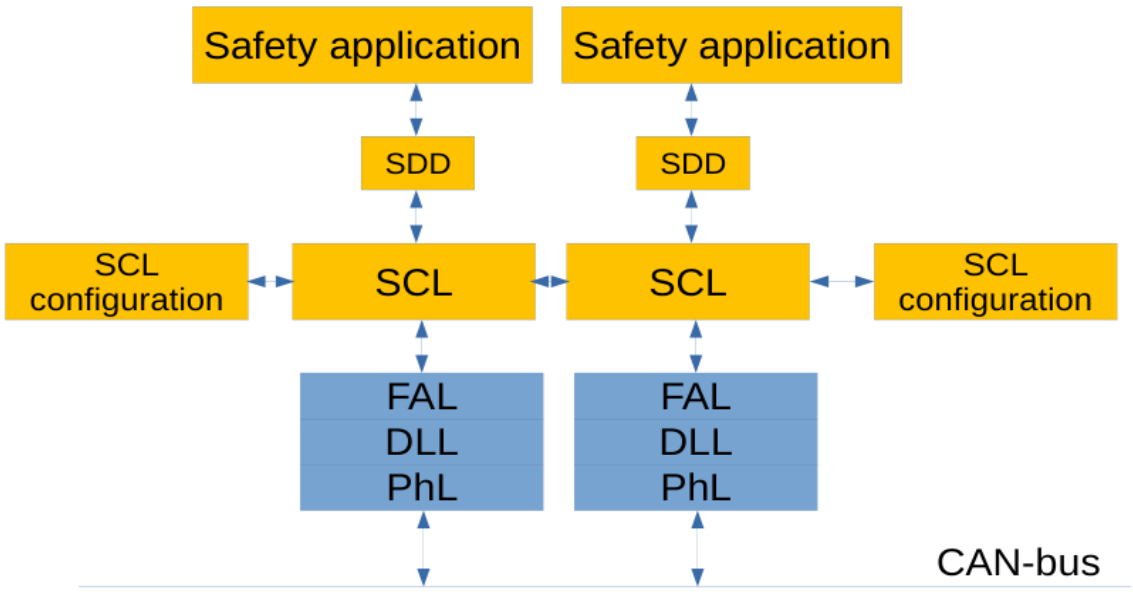

Figure 3: Example SCL structure using 61784-3:2021 Annex A Model C (A.4). SR layers are shown in orange and NSR layers are shown in blue. Note that using this model only one message is sent.

5.5 Relationships with FAL

5.5.1 General

SimpleCAN shall only be used in conjunction with EN 11989-1. There are no requirements other than those defined in this standard.

5.5.2 Data types

The following data types are handled by SimpleCAN:

- boolean values (bits);
- Unsigned 8bit integers;
- Unsigned 16bit integers.

The safe state of all data is considered as 0.

6 Safety communication layer services

6.1 Introduction

This chapter defines the services provided to the SRLD by the SCL.

6.2 Safety data dictionary

The safety data dictionary (SDD) contains the SR data to be sent and received by the SCL. The SDD shall contain up to 80 entries.

Each entry in the SDD contains the following information:

Field name	**Possible values**
Type	Producer or Consumer (ro)
SR data	0-65535 (16 bits) (rw)
SR age	0-65535 (ro)
SR valid	true / false (ro)
Node hash	0h-FFFFFFFFh (ro)

Note: an SRLD may be a producer of multiple SR packets.

6.3 Safe state signal

The SCL shall be able to signal to the SRLD to enter safe state. The maximum reaction time of the SRLD entering safe state shall be defined.

6.4 Global time

The global time is a 40-bit unsigned integer absolute time stamp counting milliseconds. The global time is synchronized via time sync control packets sent by the current active master.

6.5 Node hash

Each SC-ID shall have a corresponding node hash that is generated by the SR configuration tool. The node hash is included in the CRC calculation of each SCL to sign the data. Consumers must know the node hash of the message to be able to calculate the CRC.

The node hash shall be generated using application appropriate information.

Note: The node hash is used to ensure that consumers can not accept messages which are not signed using the same node hash as the consumer. For example, for a producer that transmits the state of an E-stop, the node hash may be based on which bit corresponds with the E-stop.

If the configuration of the producer changes which bit corresponds to the E-stop, the node hash will be different, and consumers need to update their node hash for that SC-ID to correspond to the new configuration.

6.6 Safety configuration

The safety configuration shall be generated by the SR configuration tool and verified by the SRD before initializing normal operation. If the configuration is invalid the SRD shall enter safe state.

The safety configuration shall consist of:

- The SC-IDs that the SRLD transmits;
- The node hashes of the SC-IDs that the SRLD transmits;
- The SC-IDs that the SRLD listens to;
- the node hashes of the SC-IDs that the SRLD listens to;
- The timeout of the SR data that the SRLD listens to.

6.6.1 Safety configuration CRC

The SRD and the SR configuration tool shall use the CRC algorithm with the generator polynomial 04c11db7h, or another suitable CRC algorithm/polynomial. The CRC shall be calculated by the SR configuration tool and downloaded to the SRD after downloading the configuration.

See FSD351 sheet 7 for example calculations using the polynomial 04c11db7h.

7 Safety communication layer protocol

7.1 Safety PDU format

All packets are standard CAN 11-bit frames with DLC=8 (8 bytes). The meaning of the data and which CAN-IDs are used are specified below.

7.1.1 Memory packets

Memory packets are transmitted by producers cyclically according to the network cycle time. In one network cycle every producer transmits each of their SC-IDs once. Producers transmit packets in their configured slot index.

The memory packet is the only packet type containing SR data. It is a CAN 2.0A frame which includes 16 bit safe data, and 16 bit non-safe data:

- 11 bit SC-ID (SR), (30h-7Fh).
  Note: this is the CAN-ID as defined in ISO 11989.
- 16 bits safe memories (SR)
- 16 bits non-safe memories (NSR)
- 24 bit safety CRC (SR)
- 6 bit timestamp hint: lower 6 bits of global time at packet creation. Receivers can use this to calculate the original timestamp. (NSR)
- 2 bit flags. (NSR)
- 16 bit CAN CRC (built in/made by HW) (NSR)

Flags definition:

- bit 0: if set, means this packet is sent by an SRLD that is acting as master.
- Bit 1: reserved, shall be 0.

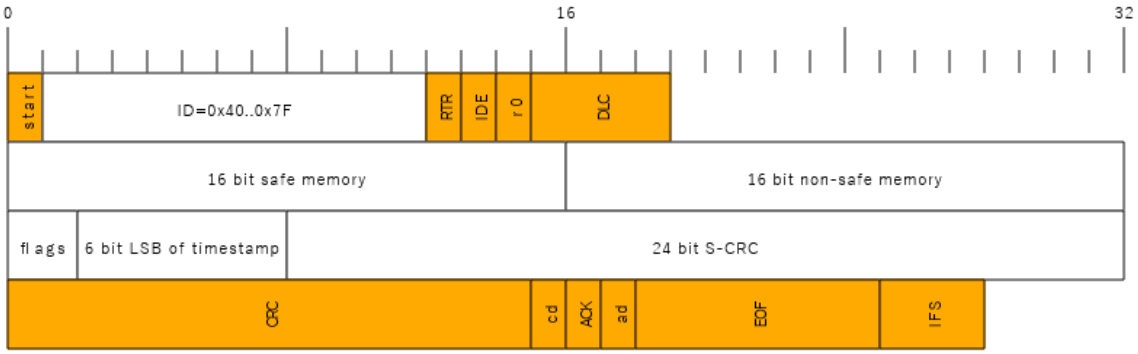

Figure 4: Memory Packet diagram (unrelated CAN-standard fields are marked orange).

7.1.1.1 Memory packet 24-bit safety CRC

An implicit mechanism (as defined in 61784-3:2021 subclause 5.8.5.1) is used for checksum calculation. The assumed equivalent knowledge consists of:

- The 40-bit global time;
- The transmitting node hash.

A 24-bit CRC algorithm is used to generate the 24-bit safety checksum for memory packets. The polynomial `CBA785h` is used. The CRC is calculated in one of the SCLs of the SRLD. The CRC is seeded with a 32-bit CRC generated by the other SCL. The 32-bit CRC polynomial shall be `04C11DB7h`.

Data structure for seed calculation:

```
// struct SimpleCAN_CRCData_CPU2
struct PACKED SimpleCAN_CRCData_CPU2
{
    uint8_t sc_id_inv;              // 0
    uint8_t counter_upper8_inv;     // 1
    uint16_t safe_mems_inv;         // 2
    uint32_t counter_lower32_inv;   // 4
}; // size=8, align=1
static_assert(sizeof(struct SimpleCAN_CRCData_CPU2) == 8, "Size of SimpleCAN_CRCData_CPU2 not as expected");
```

Example calculation of 32bit CRC:

```
/// @return
static inline uint32_t cpu2_crc_seed(uint8_t id, uint64_t timestamp, uint16_t safe_mems, uint32_t node_hash)
{
    struct SimpleCAN_CRCData_CPU2 crc_data = {
        .sc_id_inv = ~id,
        .counter_upper8_inv = ~(uint8_t)(timestamp >> 32),
        .safe_mems_inv = ~safe_mems,
        .counter_lower32_inv = ~((uint32_t)timestamp),
    };
    return (simplecan_seed_crc32_update(0, &crc_data, sizeof(struct SimpleCAN_CRCData_CPU2)) ^ (~node_hash));
}
```

The result from the CRC32 calculation is XORed with the inverted node hash, and the result of the XOR operation is the CPU2 seed.

Data structure for final calculation (CPU1):

```
struct PACKED SimpleCAN_CRCData
{
    uint8_t sc_id;                  // 0
    uint8_t counter_upper8;         // 1
    uint16_t safe_mems;             // 2
    uint32_t counter_lower32;       // 4
    uint32_t cpu2_seed;             // 8
}; // size=12, align=1
static_assert(sizeof(struct SimpleCAN_CRCData) == 12, "Size of SimpleCAN_CRCData not as expected");
```

Example calculation of 24-bit CRC:

```
uint32_t node_hash = state->config.network.node_hashes[state->prepared_tx_packet.sc_id - state->config.node.id_range_start_offset];
uint64_t timestamp = state->tick_counter - 1; // These memories are 1 tick old already (we just increased tick_counter)
uint8_t tx_id = state->precomputed.tx_ids[state->tx_pkt_index];
uint8_t memory_index = tx_id - state->config.node.id_range_start_offset;
state->prepared_tx_packet.safe_mems = simplecan_cfg_get_memory(memory_index);
state->prepared_tx_packet.counter_upper8 = (uint8_t)((timestamp >> 32) & 0xFF);
state->prepared_tx_packet.sc_id = tx_id;
state->prepared_tx_packet.counter_lower32 = (uint32_t)(timestamp & 0xFFFFFFFF);
state->prepared_tx_packet.cpu2_seed = state->seed_from_cpu2;
uint32_t safe_crc24 = simplecan_crc24_update(0, &state->prepared_tx_packet, sizeof(state->prepared_tx_packet)) ^ node_hash;
```

The result (safe_crc24) is the final 24-bit CRC.

7.2 Time sync control packets

The time sync control packet are NSR packets sent periodically from the SCL that is the current active master to synchronize the clocks in SCLs in the network. It contains the lower 40 bits of the active masters global time.

- 11 bit CAN-ID. Always 10h;
- 8 bit ctrltype. Always 0 for Time Sync;
- 8 bits *lowest* SC-ID of sender. 30h-7Fh;
- 8 bit which slot this pkt is sent in;
- 40 bits global time;
- 15 bit CAN CRC (built in/made by HW).

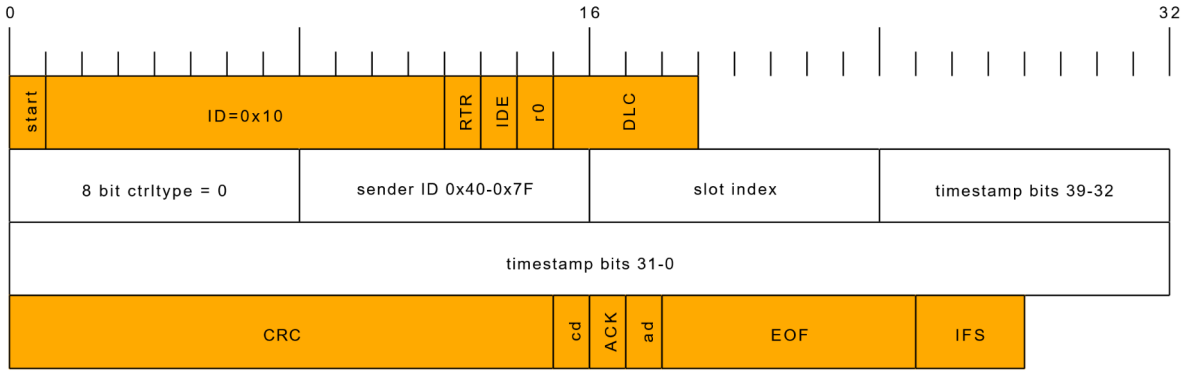

Figure 5: Time sync packet structure.

7.3 Timing requirements

The network cycle is split into slots, where each slot is 1 millisecond. Producers shall transmit their SR data in their configured slot index.

Consumers shall keep track of the age of the SR data. If the age of the data exceeds the configured safety timeout, the data shall be set to safe state (0).

The network cycle shall have at least two empty slots in the end of the cycle where no producers are configured to transmit. If a master wants to transmit a time sync packet, it shall be sent in the last slot of the cycle.

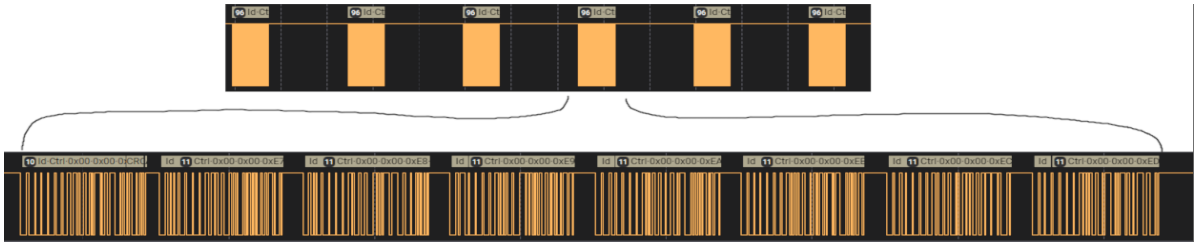

Figure 6: Timing diagram of a SimpleCAN network, showing 6 cycles with 8 memory packets.

7.3.1 Active master time sync control packet transmission

The SCL that is currently acting as master periodically sends time sync control packets containing the current global time. It sends it to all SCLs in the network (including the SCL in the same SRLD).

The active master shall not send time sync packets more often than once per 100ms.
The active master shall not send time sync packets less often than once per 500ms.

The active master SCL shall verify that the time sync packet has been successfully transmitted on the bus at most 2ms after transmitting the packet. If the transmission has failed or not started after 2ms (for example due to CAN-arbitration or other external errors), the transmission shall be aborted and any buffers cleared.

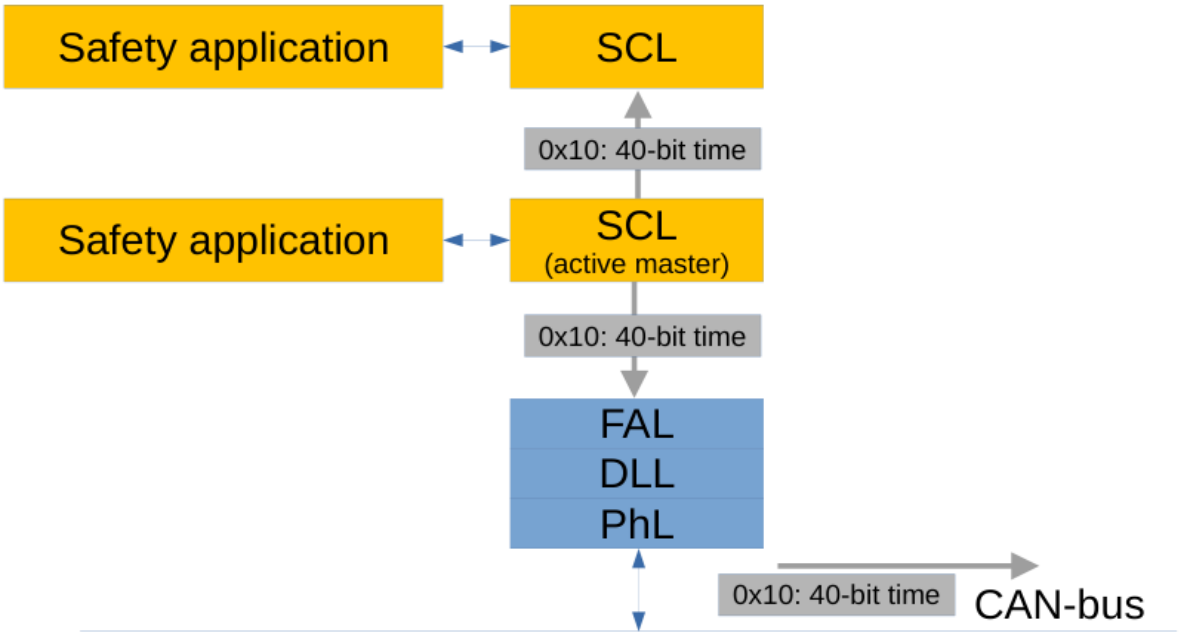

Figure 7: The propagation of 0x10 time sync packets from the SCL acting as the active master. The master sends the time sync packet on the CANbus, and to the other SCL of the SRLD.

7.3.2 Memory packet transmission

Memory packets are transmitted by one of the SCLs. The 24-bit CRC is based on the 32-bit seed from the other SCL.

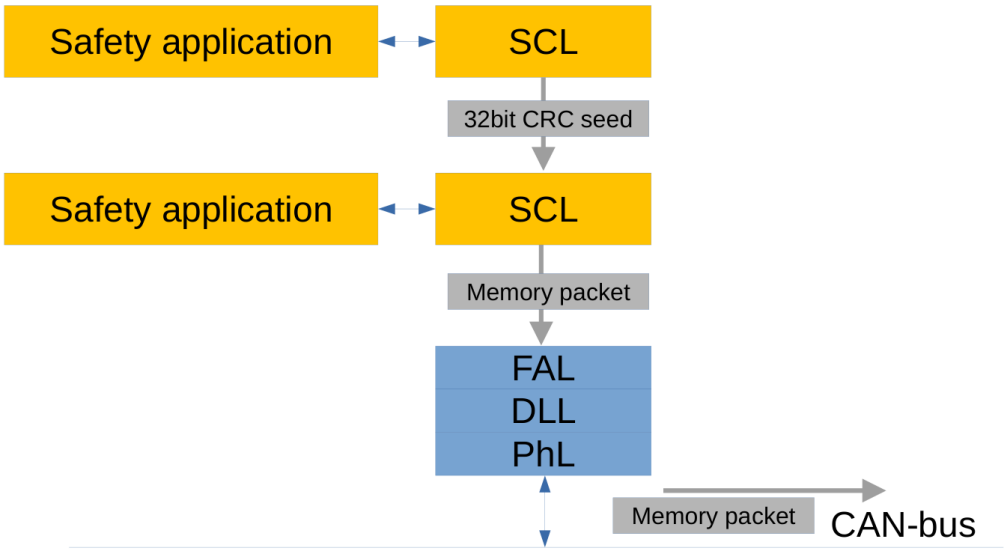

Figure 8: Flow of information during memory packet transmission.

7.3.3 Memory packet reception

Memory packets are received by SCLs connected to the bus. If only one SCL is connected to the bus (model A), the SCL shall forward the packet to the other SCL without unpacking (CAT3, HFT=1).

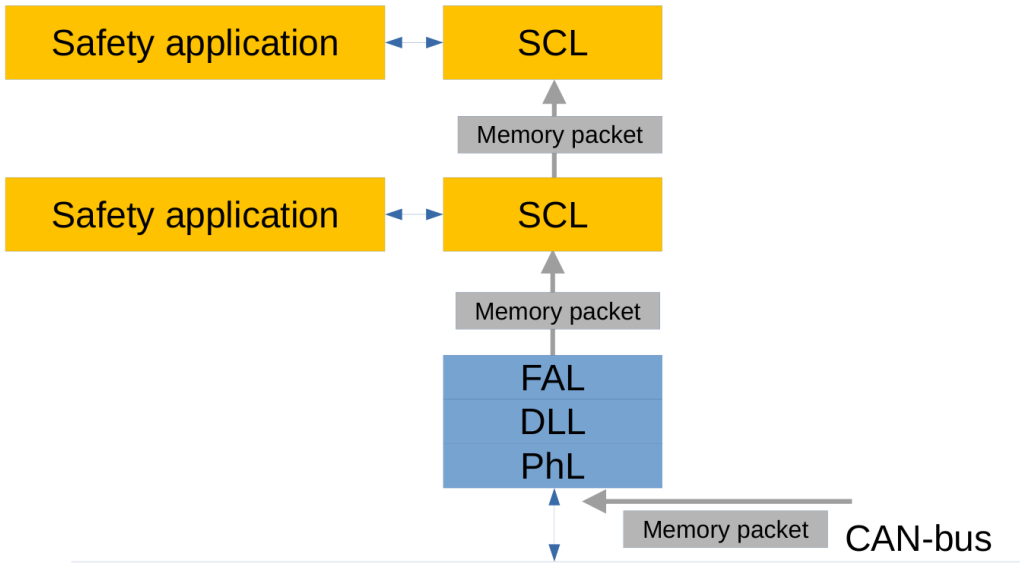

Figure 9: Flow of information during memory packet reception.

7.4 SR network initialization and system boot-up

The potential masters of the SR network wait a specified amount of time, and then start transmitting time sync packets if no master with higher SC-ID is heard (see 8.2).

7.4.1 Determining the active master

There shall only be one active master on the bus at a given time. If multiple SCLs can act as master, the SCL with the lowest transmitted SC-ID shall take the role of master. If a potential master joins the network late, before taking over the role as active master, the potential master shall first synchronize its global time to the current network global time.

If an SCL on the bus detects multiple active masters it shall signal to the SRLD to enter safe state.

The formula below specifies how long the masters shall wait based on their lowest transmitted SC-ID:

T_{wait} = (SC-ID – 0x30) * 5 [ms]

This is to avoid collisions on the bus for the first time sync packet.

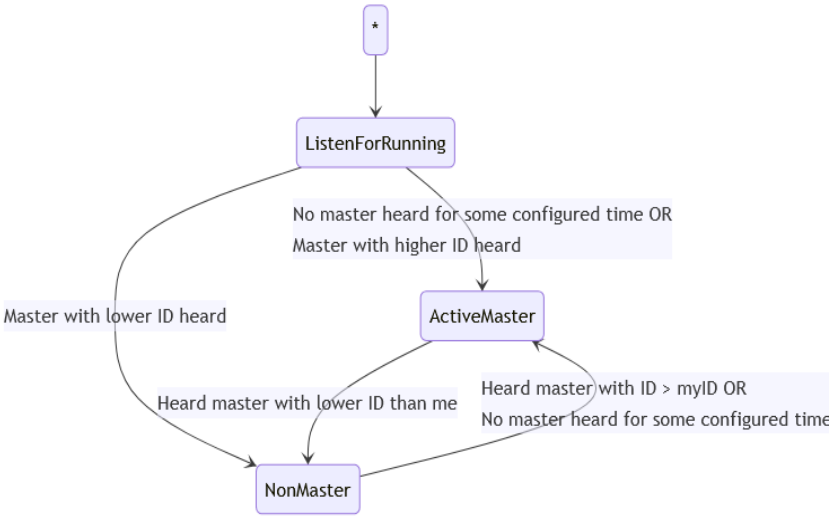

Figure 10: State diagram explaining master determination.

If any CAN error occurs during transmission of the time sync packet (for example, in case two masters try to transmit sync packets at the same time and a collision occurs), they shall back off and try again after T_{wait} ms have passed.

When the active master has been determined, the master can start transmitting time sync packets and the SCLs in the network can synchronize and start to transmit and receive SR data.

When the active master has queued a time sync packet in hardware buffers to be sent, it shall verify that the packet has been fully sent after at most 2ms. If it has not been fully sent, the transmission shall be aborted and the buffers cleared.

If more than half of time sync packets are not sent within 2ms during 10 seconds, the SCL shall signal to the SRLD to enter safe state.

7.4.2 Time synchronization

Two states for producers and consumers are defined:

- Unsynced
- Synced

All SCLs shall start in the unsynced state. In the unsynced state, producers shall not transmit any SR data, and consumers shall discard all received SR data. In the synced state, producers may transmit SR data, and consumers may handle received SR data.

To go from unsynced to synced state, two valid time sync packets from the same master shall be received. The source SC-ID of the time sync packet can be determined by looking at the 8bits SC-ID field in the time sync packet. A time sync packet is valid if the time between the previous time sync packet and the latest time sync packet is within +-2ms of the receiver SCL internal time. If the last two time sync packets are within +-2ms of the receiver SCL internal time, the receiver SCL shall accept the time as the global time and go to the synced state.

The probability of two consecutive corrupt time sync packets undetected by the 15-bit HW CRC is equal to 1.7×10^{-10} (see FSD351 page 6).

Clocks in SRLDs shall have a maximum inaccuracy of 50ppm.

If no time sync is received for 2000ms, the SCL shall go to unsynced mode.

Receivers shall specify the guaranteed maximum time a received packet can be buffered for, and add this time to the age all received time sync packets when calculating the global time.

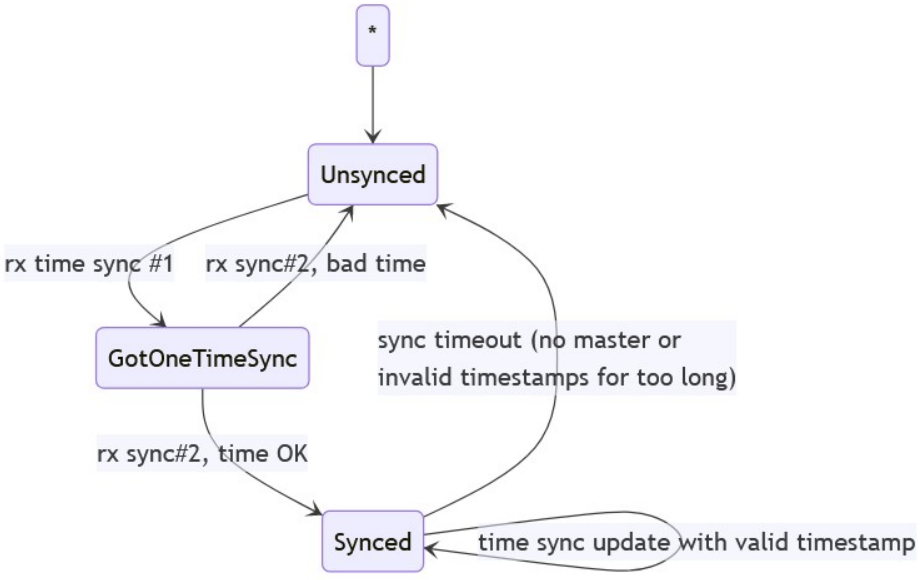

Figure 11: State diagram explaining time synchronization in SCLs.

7.5 CAN parameters

Acceptable CAN baudrates are 125kbit/s, 250kbit/s, 500kbit/s, and 1000kbit/s (1Mbit/s).

7.5.1 CAN-ID Ranges

The following CAN-IDs are defined:

- 10h Control packet (Time sync)
- 30h-7Fh SC-IDs, memory packets.
- 80h-BFh SimpleCAN diagnostic information packets.
- 130h-13Fh Simplifier standard debug data packets.

8 Safety communication layer management

8.1 SR device configuration

The SRD shall perform the SR device configuration verification before entering normal operation. The SR device shall calculate a CRC signature as defined in 6.6.1. The calculated CRC signature shall be compared with the safety configuration signature (see 6.6.1). If both values are equal the configuration shall be valid.

8.1.1 SR configuration tool

The SR configuration tool shall perform the configuration download to the SR devices in the network. The user is responsible for correctly addressing the SR devices on the network during configuration download. The safety manual of the SR devices shall contain instructions on how the user can achieve this (see 9.6). The SR configuration tool shall have measures to help the user correctly address the SR devices. Residual error rate RR_{CP} is given in FSD351 page 7.

8.2 SR network configuration

The methods and algorithms required to verify the validity of the SR network configuration do not fall into the scope of SimpleCAN. The user is responsible for correct parameterization of the network.

8.3 Communication phases

Setup or change of the SCP configuration in an SRLD shall only be possible if the SRLD is in safe state. No safety communication shall be possible in safe state. Warm start after fault is only possible with a complete reset and initialization of the SRLD.

If the safe 24-bit CRC is invalid for 1000 packets in one hour, the SCL shall signal to the SRLD to enter safe state. An algorithm to achieve this is suggested:

- A counter is stored in memory counting the number of invalid CRCs.
- If a packet with an invalid CRC is received, the counter is increased by 1.
- Every 3600/1000≈3.6 seconds, if the counter is not equal to 0, this counter is decreased by 1.
- If the counter reaches 1000, the SCL signals to the SRLD to enter safe state.

8.4 Security considerations

Unauthorized access to the SRLDs in the network shall be managed by the SRLD implementation.

Since CAN is based on physical wires, access to the CAN network requires direct physical access to the wires. Tampering with the CAN network is equivalent to directly short circuiting safety devices or machines, and thus is not in the scope of SimpleCAN.

9 System requirements

9.1 Indicators and switches

Indicators and switches are depending on the individual SRD and is not in the scope of SimpleCAN.

9.2 Installation guidelines

SimpleCAN shall only be used with ISO 11989. The PhL shall be continuous without any devices separating SRLDs on a network. Appropriate standards shall be considered depending on the application field. In machinery and process environment the principles defined in the common part of EN 61918 shall apply.

Only SimpleCAN compatible devices can be transmitters in a SimpleCAN network. Silent listeners are allowed to listen to the network in silent mode (according to CAN2.0).

These requirements shall be explained to the user in the manual.

9.3 Safety function response time

The safety reaction time of SimpleCAN shall be the application configured safety timeout. This timeout shall account for all components of the SCL.

9.4 Constraints for the calculation of system characteristics

9.4.1 Number of memory packets

The number of memory packets per hour is 3600*4000.

The number of invalid memory packets (CRC24 div != 0) allowed per hour is 1000.

The maximum rate of time-sync telegrams per hour is 3600*10.

9.4.2 Residual error rate calculations

See FSD351 for detailed calculations. The total residual error rate for the FSCP is given by

$\lambda_{SC} = RR_T + RR_A + RR_I + RR_M$ Which given the calculations below is equal to

$\lambda_{SC} = 6.24\text{E-}12 + 7.80\text{E-}12 + 3.64\text{E-}12 + 6.25\text{E-}27$ = **1.79E-11**

9.4.2.1 Contribution of data integrity errors (RR_I)

The residual error rate for Data Integrity RRI is given by the formula $RR_I = RP_I \times v \times RP_{FSCP\_I}$

RP_I is given in FSD351 sheet 5 as 6.24E-15, v is the maximum number of packets per hour, 3600*4000 = 1.44E7, RP_{FSCP}=1.00E3/1.44E7, the maximum number of bad packets allowed per hour. This gives:

$$RR_I = 6.24\text{E-}15 \times 1.44\text{E}7 \times \frac{1.00\text{E}3}{1.44\text{E}7} = \mathbf{6.24E\text{-}12}$$

9.4.2.2 Contribution of authenticity errors (RR_A)

The error rate of authenticity errors (RR_A) is determined by the residual error probability for data integrity (RP_I), the number of bits in the A-code (8 transmitted in the packet are included in RP_I, 32 are implicit). Of 2048 possible CAN-IDs, only 80 are valid for safety data, and the number of invalid packets allowed per hour (1000). This gives:

$$RR_A = RP_I \times \frac{80}{2048} \times 32 \times 1.00\text{E}3 = \mathbf{7.80E\text{-}12}$$

9.4.2.3 Contribution of timeliness errors (RR_T)

The formula $RR_T = 2^{-LT} \times w \times R_T \times RP_{FSCP\_T}$ gives the contribution of timeliness errors.

LT=40 (Global time counter), w=2, since only the timestamp with the 6 corresponding lower bits is a valid timestamp, with potentially one adjustment to bits 7-40. R_T is 2E-3 × 1E3, since the sender and receiver have each one storing element (5.8.8.2), and up to 1000 bad packets are allowed. This gives:

$$RR_T = 2^{-40} \times 2 \times 2\text{E-}3 \times 1000 = \mathbf{3.64E\text{-}12}$$

9.4.2.4 Contribution of masquerade errors (RR_M)

The contribution of masquerade errors can be calculated by the number of devices, the rate of masquerade errors per device, the allowed range of values for the CAN-ID field, and the 24-bit safety CRC:

$$RRM = 80 \times 1.00\text{E-}3 \times \frac{80}{2048} \times 2^{-24} = \mathbf{6.25E\text{-}27}$$

9.5 Maintenance

There are no special maintenance requirements for this protocol.

9.6 Safety manual

Implementers of SimpleCAN shall supply a safety manual with the following information at a minimum:

- Installation guidelines (see 9.2);
- the safety manual shall inform the users of constraints for calculation of system characteristics (see 9.4);
- the safety manual shall inform the users of their responsibilities in the proper parameterization of the devices (8.2.1);
- the safety manual shall contain advises on calculating the expected maximum network reaction time.

In addition to the requirements of this clause the safety manual shall follow all requirements in the EN 61508 series.